

ERROR 404

سوال ۹ - گره های لیست

به نام خدایی که از نسبت محیط

دایره به قطر آن آگاه است

غیاث الدین جمشید کاشانی

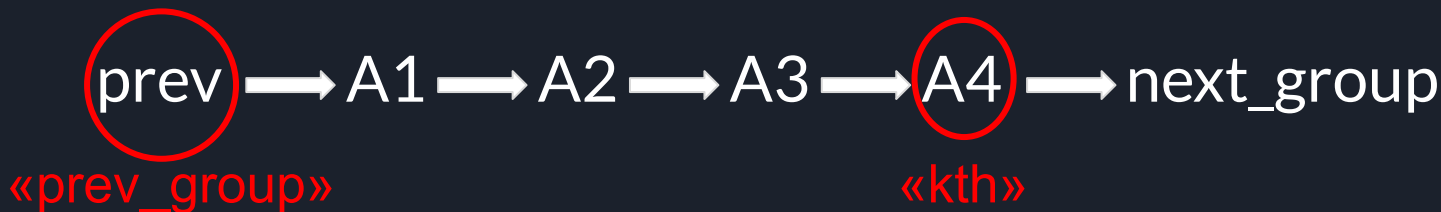
- **امیرعلی نیافر**
- **محمد هادی نجفی**
- **محمد علی ارجمند لاری**
- **محمد سبحان بدخشانی**

استان قم - پایه دهم
دبیرستان استعدادهای درخشان شهید قدوسی

توجه کنید در ابتدا فرض می‌کنیم دسته k تایی مورد نظر ما اولین دسته موجود در لیست نیست! (بنابر این متغیر «prev_group» از قبل تعریف شده و برابر آخرین عضو دسته k تایی قبلی است. یعنی عضوی که اتریبیوت next اش به اولین عضو دسته k تایی هدف اشاره می‌کند)

با یک حلقه while و شمارنده «kth» از «prev_group» شروع کرده و وجود دسته k تایی پیش رو را چک می‌کنیم. بدین صورت پس از اتمام حلقه و در صورت وجود دسته k تایی، متغیر «kth» آخرین عضو آن دسته k تایی را در لیست اولیه نشان می‌دهد.

در ادامه برای فهم بهتر الگوریتم یه مثال با $k = 4$ می‌زنیم



در ادامه k بار و به ازای هر عضو از دسته k تایی هدف با تعریف سه متغیر، اتریبیوت $next$ هر گره را تغییر می‌دهیم تا دسته معکوس شود

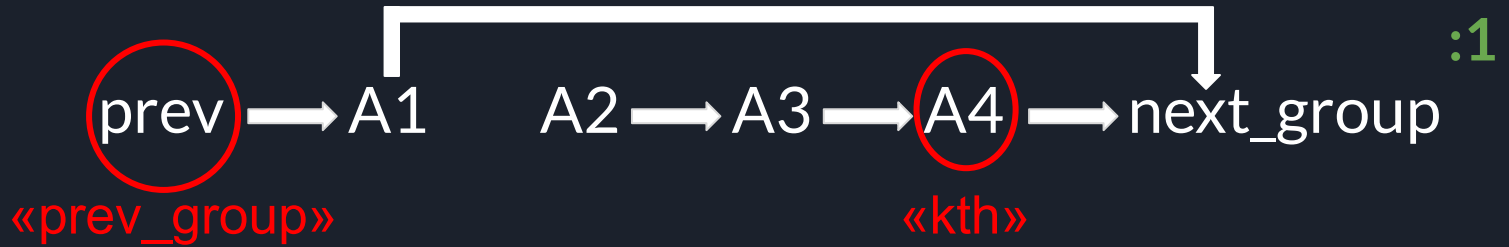
سوال ۹ - گره هاوی لیست
ERROR 404

$curr$: گرهی که در مرحله موجود در آن قصد تغییر اتریبیوت $next$ اش را داریم.
 $dist$: گرهی که قرار است مقدار جدید اتریبیوت $next$ از گره $curr$ باشد.
 nxt : حاوی مقدار اتریبیوت $next$ از گره $curr$ قبل از تغییر این اتریبیوت به $dist$

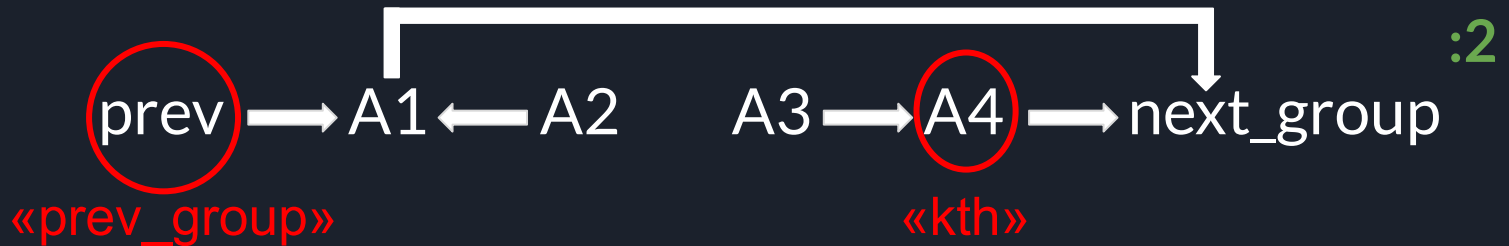
و در هر مرحله ابتدا مقدار اتریبیوت $next$ از گره $curr$ را در nxt ذخیره می‌کنیم. سپس مقدار اتریبیوت $next$ از گره $curr$ را برابر $dist$ قرار می‌دهیم. و برای آماده سازی مرحله بعدی، $dist$ را برابر $curr$ قرار می‌دهیم. همچنین در مرحله بعد قرار است گره هدف همین گره nxt باشد. پس در ادامه $curr$ را برابر nxt می‌کنیم (دلیل ذخیره کردن اتریبیوت $next$ از $curr$ قبل از تغییر همین بود. زیرا در مرحله بعدی قرار است تغییرات روی آن مقدار انجام شود).

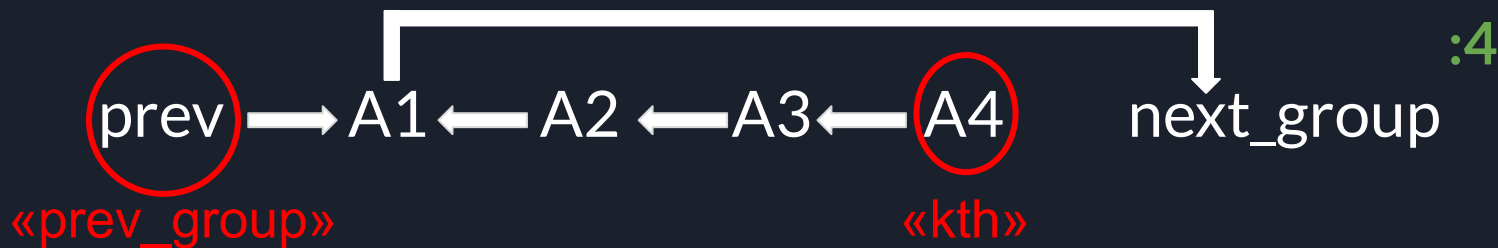
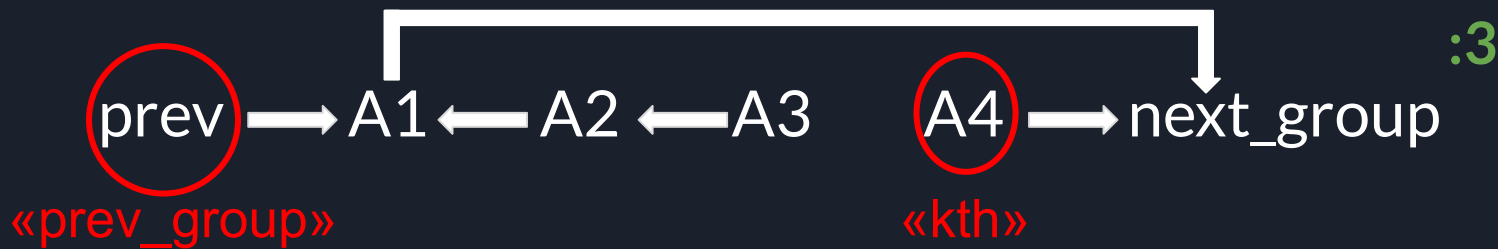
در اولین مرحله از این بخش از الگوریتم، می‌خواهیم گره $A1$ را به دسته بعدی وصل کنیم. بنابراین در گام اول $curr=A1$ و $dist=kth.next$ (بدیهی است $kth.next$ به اولین عضو از دسته بعدی اشاره می‌کند)

بعد از انجام اولین مرحله نتیجه به صورت زیر است:

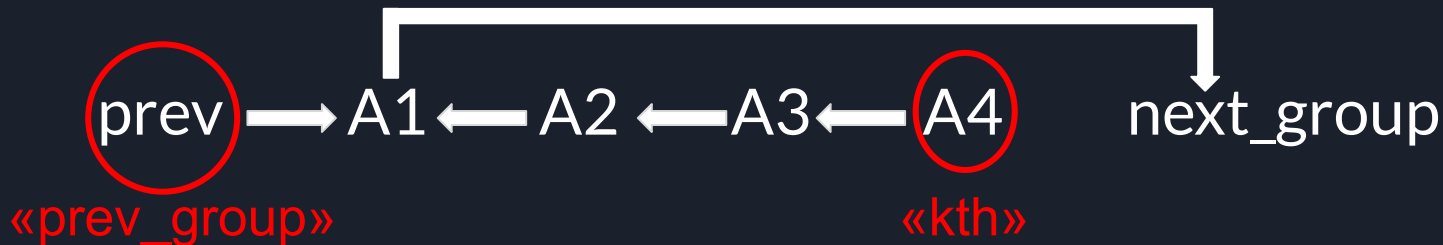


اگر طبق توضیحاتی که در گذشته برای هر مرحله داده شد پیش برویم نتایج بعد از هر مرحله به ترتیب طبق نمودار های زیر است (k بار این مراحل را تکرار کنیم):

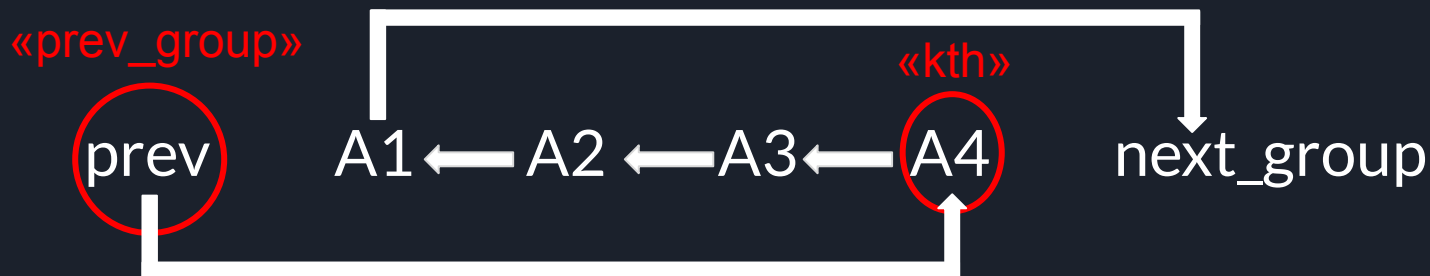


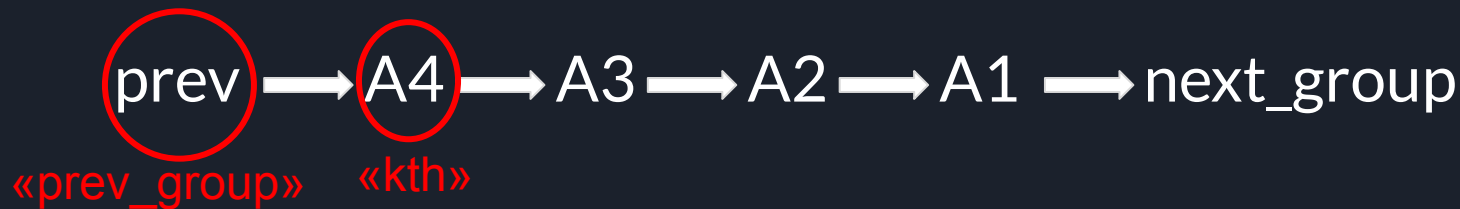


حال مشاهده کردیم که دسته k تایی هدف، به صورت کاملاً صحیح معکوس شد.
اما به وضوح قابل مشاهده است که این دسته در اتصال با کل لیست مشکل دارد:



بنابراین می بایست آخرین عضو دسته قبلی («prev_group») را به اولین عضو دسته هدفمان در چینش جدید (که همان آخرین عضو دسته هدفمان در لیست اولیه بود) وصل کنیم.
این گروه همان متغیر «kth» است. پس می نویسم: $prev_group.next = kth$





تقریباً کار به اتمام رسیده. اما نکته مهم اینکه برای شروع کار در دسته بعدی باید در همین دور «prev_group» را به روز کرد و آن را برابر آخرین عضو دسته هدف در چینش جدید (که همان اولین عضو دسته هدف در لیست اولیه بوده) کرد که این گره در واقع همان مقدار لیست که «prev_group» قبل از تغییر مقدار اتریبیوت next اش به آن اشاره می کرد.

بنابراین به عقب بر می گردیم و قبل از اینکه مقدار next از گره «prev_group» برابر «kth» قرار دهیم، آن را در متغیر «nxt» ذخیره می کنیم و بعد از تغییر مقدار next از گره «prev_group» به «kth»، خود متغیر «prev_group» را برابر next قرار می دهیم.

یعنی:

• `nxt = prev_group.next`

• `prev_group.next = kth`

• `prev_group = nxt`

سپس به سراغ دسته k تایی بعدی می‌رویم.

در اسلاید اول ذکر شد فرض می‌کنیم دسته k تایی هدفی که درباره آن صحبت می‌کنیم اولین دسته k تایی موجود در لیست نیست. به این دلیل که اولین دسته k تایی در لیست، دسته قبلی ندارد بنابراین تعریف متغیر «`prev_group`» برای اولین دسته به مشکل می‌خورد! برای حل این مشکل و برای اینکه از شرط خاصی برای دسته اول استفاده نکنیم (که باعث پیچیدگی الگوریتم شود)، در ابتدای الگوریتم یک گره کاذب به نام «`dummy`» ایجاد کرده و اتریبیوت `next` آن را برابر `head` لینک لیست قرار می‌دهیم. به این شکل حالا اولین دسته k تایی نه به مشکل می‌خورد و نه شرط خاصی می‌خواهد.

در ابتدای برنامه، کلاسی برای ساخت سیستم لینک لیست تعریف شده که برای هر گره دو اتریبیوت « 1- مقدار و 2- بعدی (شی گره بعدی) » را نگه می‌دارد.

در انتهای برنامه هم دو تابع وجود دارد:

1. ورودی ساده کاربر را به لینک لیست تبدیل کند
2. لینک لیست خروجی را به لیست قابل مشاهده تبدیل کند

بنابراین فرمت ورودی به شکل زیر می‌باشد

مثال ورودی	فرمت ورودی	شماره خط
3	عدد k	1
1 2 3 4 5 6 7 8	object 1 object 2 object 3 ...	2
['3', '2', '1', '6', '5', '4', '7', '8']	خروجی مثال	